

PC2Model: ISPRS benchmark on 3D point cloud to model registration

Mehdi Maboudi¹, Said Harb¹, Jackson Ferrao¹, Kourosh Khoshelham², Yelda Turkan³, Karam Mawas¹

¹ Institute of Geodesy and Photogrammetry, Technische Universität Braunschweig, Germany
(m.maboudi, s.harb, j.ferrao, k.mawas)@tu-braunschweig.de

² Department of Infrastructure Engineering, University of Melbourne, Australia. k.khoshelham@unimelb.edu.au

³ Civil & Construction Engineering, Oregon State University, USA. yelda.turkan@oregonstate.edu

Keywords: Dataset, 3D Model, Simulation, Point Cloud, Co-registration, LiDAR.

Abstract

Point cloud registration involves aligning one point cloud with another or with a three-dimensional (3D) model, enabling the integration of multimodal data into a unified representation. This is essential in applications such as construction monitoring, autonomous driving, robotics, and virtual or augmented reality (VR/AR). With the increasing accessibility of point cloud acquisition technologies, such as Light Detection and Ranging (LiDAR) and structured light scanning, along with recent advances in deep learning, the research focus has increasingly shifted towards downstream tasks, particularly point cloud-to-model (PC2Model) registration. While data-driven methods aim to automate this process, they struggle with sparsity, noise, clutter, and occlusions in real-world scans, which limit their performance. To address these challenges, this paper introduces the PC2Model benchmark, a publicly available dataset designed to support the training and evaluation of both classical and data-driven methods. Developed under the leadership of ICWG II/Ib, the PC2Model benchmark adopts a hybrid design that combines simulated point clouds with, in some cases, real-world scans and their corresponding 3D models. Simulated data provide precise ground truth and controlled conditions, while real-world data introduce sensor and environmental artefacts. This design supports robust training and evaluation across domains and enables the systematic analysis of model transferability from simulated to real-world scenarios. The dataset is publicly accessible at: <https://zenodo.org/records/17581812>.

1. Introduction

The proliferation of three-dimensional (3D) sensing technologies has fundamentally transformed how physical environments are captured and analysed. Devices such as Light Detection and Ranging (LiDAR) scanners, photogrammetric systems, and structured light sensors now produce dense geometric representations of real-world scenes at scale, and their growing affordability has made such data commonplace across disciplines (Maboudi et al., 2025). In fields ranging from construction site monitoring and autonomous driving to robotics, digital 3D printing, and virtual and augmented reality (VR/AR), there is a pressing need to interpret these representations in relation to prior geometric knowledge, most often in the form of 3D models (Huang et al., 2024, Mora et al., 2021). This has driven continued interest in point cloud registration: the task of determining the spatial transformation that best aligns a captured point cloud with a reference, whether that reference is another

point cloud or a 3D model. While point cloud-to-point cloud registration has been studied extensively and is supported by a variety of established benchmarks, the analogous problem of point cloud-to-model registration remains relatively underexplored. Existing datasets do not generally accommodate this setting, limiting the ability to train and evaluate data-driven approaches. To address this, we introduce the ISPRS benchmark on 3D point cloud-to-model registration, PC2Model. It features 3D models and their transformed scans for alignment (co-registration). Figure 1 illustrates the co-registration process schematically.

Algorithms for point cloud registration encompass a broad range of techniques, from established methods such as iterative closest point (ICP) to deep learning-based approaches (Zhao et al., 2022). In practice, point clouds obtained from real-world scans present challenges such as sparsity, clutter, noise, mixed pixels, and occlusions, which

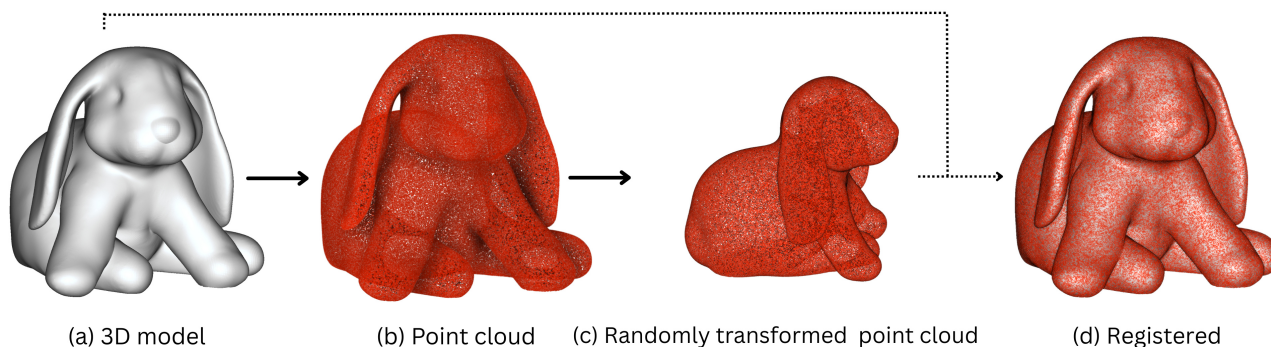


Figure 1. Point cloud simulation from 3D model and its co-registration. a) 3D model, b) point cloud sampled from 3D model, c) point cloud after arbitrary 7-parameter transformation, and d) transformed point cloud registered to the 3D model.

reduce the performance of point cloud registration algorithms. The PC2Model benchmark includes these artefacts in a controlled and systematically reduced manner, providing a dataset that closely approximates real-world conditions. Developed under the leadership of ICWG II/I b, PC2Model provides a publicly accessible dataset for 3D point cloud-to-model registration, encompassing 137 samples. A key feature of the dataset is its hybrid design, combining simulated point clouds, and, in some cases, real-world scans with their corresponding 3D models. The simulated point clouds offer precise ground truth alignments and controlled acquisition conditions, whereas the real-world data introduce sensor- and environmentally induced artefacts such as beam divergence, noise, and measurement inaccuracies. This hybrid configuration facilitates the training and evaluation of data-driven registration methods across both domains and enables systematic investigation of the transferability of models trained on simulated data to real-world scenarios.

The rest of this paper is organised as follows: In section 2, we review point cloud-to-point cloud and point cloud-to-mesh co-registration methods, and discuss existing datasets relevant to point cloud-to-model co-registration. Section 3 describes the PC2Model dataset and includes the categories of objects and details of the point cloud simulation of the 3D models. Section 4 describes the simulation environment and the implemented add-on. Section 5 then presents and discusses evaluation strategies and metrics. Finally, section 6 summarises the paper and outlines directions for future work.

2. Related Work

The registration of point clouds to 3D models is a fundamental problem in computer vision, robotics, and photogrammetry. Over the years, a wide range of classical and learning-based methods have been developed, each with its own strengths and limitations. This section reviews representative approaches, spanning conventional techniques to modern learning-based methods.

Conventional methods: The Iterative Closest Point (ICP) algorithm (Besl and McKay, 1992) is one of the most widely used methods for rigid registration. It iteratively establishes correspondences between a source and a target (point cloud or model), and computes the rigid transformation that minimises the distances between matched points. ICP performs well when a good initial coarse alignment is available; however, it is sensitive to local minima, symmetry ambiguities, or insufficient geometric features (Mawas et al., 2026). To address these limitations of naive ICP, Go-ICP (Yang et al., 2016) employs a branch-and-bound strategy to find the optimality of the rigid transformation. The Normal Distributions Transform (NDT) (Biber and Strasser, 2003) adopts a probabilistic approach by representing the target cloud as a set of Gaussian distributions over discretised voxels and aligning the source points by maximising the likelihood. Unlike ICP, NDT does not rely on explicit point-to-point correspondences, improving robustness to variations in point density, noise, and partial overlap. Fast Global Registration (FGR) (Zhou et al., 2016) provides an efficient alternative to sampling-based approaches by directly optimising a robust objective over a dense set of candidate correspondences derived from Fast Point Feature Histograms (FPFH). Using a scaled Geman-McClure estimator with graduated non-convexity, FGR suppresses outliers without requiring iterative closest-point queries, achieving accuracy com-

parable to well-initialised local refinement methods such as ICP, while offering significant computational gains.

Learning-based methods: To tackle the co-registration problem, learning-based methods have been proposed. Deep Closest Point (DCP) (Wang and Solomon, 2019) learns pointwise features from source and target clouds and predicts the rigid transformation via a differentiable SVD (singular value decomposition) layer. PointNetLK (Sarode et al., 2019) combines the PointNet architecture with the classic Lucas–Kanade alignment method, extracting global features, and iteratively updating the transformation to minimise feature differences. These methods demonstrate how feature learning can improve robustness and generalisation across a wide range of point cloud shapes and densities. In summary, classic algorithms such as ICP, Go-ICP, and NDT remain foundational for rigid registration, while learning-based approaches such as DCP and PointNetLK show promise in handling complex and diverse point cloud scenarios.

Point cloud-to-model registration: In contrast to the prior-mentioned studies that use point clouds as reference data, using a mesh as a reference can improve robustness by explicitly enforcing visibility (line-of-sight) constraints during correspondence selection. In this framework, synthetic range images are rendered from the meshes and integrated into a Monte Carlo Localisation (MCL) pipeline (Chen et al., 2021). In contrast, ICP-like methods have incorporated ray-casting correspondences (RCC) to register sensor data with a mesh model (Vizzo et al., 2021). RCC is introduced to reduce incorrect matches in challenging configurations such as thin structures and self-occlusions, but the presented implementation did not achieve real-time performance. RCC efficiency is substantially improved in (Mock et al., 2024) by leveraging heterogeneous parallel hardware, while estimating the rigid transformation using a standard SVD-based approach. A geometry-based approach by (Zhao et al., 2022) addresses cross-form registration between LiDAR point clouds, BIM models, and HoloLens meshes by extracting planar surfaces and matching them across data types, achieving this without manual initialisation.

Existing datasets related to point cloud to 3D model alignment mainly originate from adjacent tasks such as object pose estimation or multimodal registration. The LINEMOD (Hinterstoisser et al., 2013) and TUD-L (Hodaň et al., 2018) datasets, provided within the BOP benchmark (Hodaň et al., 2018), pair object-centric point clouds derived from depth images with corresponding CAD models and ground-truth 6D poses. These datasets primarily focus on instance-level object pose estimation in cluttered scenes and cover only a small number of object categories. LINEMOD comprises 15 objects and TUD-L just 3, making them limited in scope for benchmarking registration methods that must generalise across diverse geometries. The 3DPCD-CT dataset (Saiti and Theoharis, 2022) addresses the registration of point clouds to surface representations by providing paired point clouds and mesh-like models with known rigid transformations. While this dataset explicitly targets point cloud to model alignment, it is limited in scale and variety and focuses on a specific application domain. In contrast, PC2Model is designed as a dedicated benchmark for point cloud-to-3D-model registration at a broader scale, comprising 137 object instances that span a wide variety of categories, shapes, and scanning conditions, including varying point densities, viewpoints, and realistic scanning artefacts. Unlike existing datasets, PC2Model emphasises generalisation across object classes and challenging initial misalignments, making

Source	Category	Number of samples
Simulated from 3D model	Mechanical objects	25
	Furniture	25
	Home décor	25
	Houses	25
	Vehicles	25
	Indoor spaces	6
Real	Indoor spaces	6
		Total: 137

Table 1. Categories and number of samples. The Indoor spaces (real) are from (Khoshelham et al., 2020).

it well-suited for evaluating both classical and learning-based registration algorithms under realistic conditions.

3. Dataset

The PC2Model benchmark dataset comprises simulated and real-world point clouds divided into seven categories. The six simulated categories and one real-world category are shown in Table 1, along with the number of samples. The real category contains scans acquired in real-world indoor environments with a laser scanner. Each sample includes a 3D model, the corresponding point cloud generated by either a simulated or a real laser scanner, and a rigid transformation applied to the scan, as well as the associated transformation matrices.

The dataset statistics are presented in Table 2. Houses and indoor scenes represent the largest objects with the most complex geometry; therefore, they typically have a higher number of scanners, a larger number of points, and a higher average median point density. The object categories (mechanical objects, furniture, vehicles, and home décor) generally require fewer scanners to cover the entire object, as the objects are smaller and exhibit fewer occlusions than houses with multiple rooms and levels. Point densities for houses and indoor scenes are significantly higher than for object categories, where objects are scanned in open space. The lowest point densities are observed in furniture scans. Point density was computed by forming a sphere around each point in the point cloud and counting the number of points within the sphere. This count was then divided by the volume of the sphere, and the median of these densities is calculated. For houses, vehicles and indoor scenes, the radius can be interpreted as meters; however, this does not necessarily apply to the other categories.

Depending on the model category, different scanning strategies were used. For mechanical objects, furniture, vehicles, and

Source	Category	Avg. nr. of scans	Avg. nr. of points	Avg. median point density ($r = 0.5$ m)
Simulated from 3D model	Mechanical objects	6	191,961	5,495
	Furniture	6	221,275	67
	Home décor	5	244,270	4,881
	Houses	19	24,823,458	47,542
	Vehicles	4	277,288	19,228
	Indoor spaces	13	21,009,545	43,872
Real	Indoor spaces	N/A	19,507,569	30,127

Table 2. Dataset statistics. All scans were performed with simulated terrestrial laser scanners, except for indoor spaces (Real).

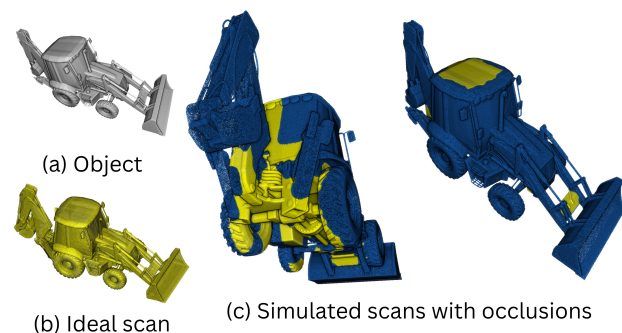


Figure 2. Occluded regions (yellow) and scanned regions (blue) compared to an ideal scan, shown from an inverted viewpoint.

home décor, scanners were placed on the ground plane surrounding the object. As a result, the upper parts of these objects may not always be fully scanned. In addition, vehicles, houses, and indoor scenes were scaled to their approximate real-world dimension. The rest of the object categories were intentionally not scaled to their real-world sizes to introduce additional size variability into the dataset. Some houses are furnished, while others contain empty rooms; however, the scanning strategy remained the same. Scanners were placed at approximately tripod height (1.5 m to 2 m) above ground level in each room, where additional areas, such as the roof spaces, were accessible, for example, through connecting doors; further scanners were placed accordingly. Furthermore, scanners were positioned outside the houses at ground level to capture the exterior walls. For indoor scenes, scanners were also placed in each room. In contrast to houses, no scanners were positioned outside indoor scenes; therefore, the exterior walls were not scanned. The real-world indoor scans are from the ISPRS benchmark for indoor modeling (Khoshelham et al., 2020) and feature six different buildings with various complexities. In Figure 6, a visual overview of samples of each category is depicted.

The PC2Model benchmark includes common real-world scanning artefacts in a controlled and systematically reduced form, approximating realistic scanning conditions without eliminating these effects entirely. Mixed pixels arise along object edges where the laser beam partially hits multiple surfaces, producing spurious points, as shown in Figure 4. By inspecting Cloud-to-Mesh distance, depicted in Figure 3, noise manifests as surface deviations from the ground-truth model, visible as localised point displacements. Occlusions result in missing regions

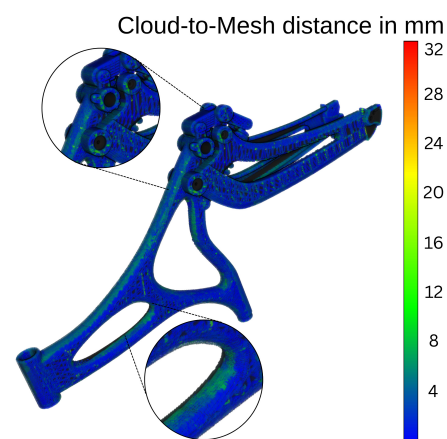


Figure 3. Surface deviation from the ground-truth model.

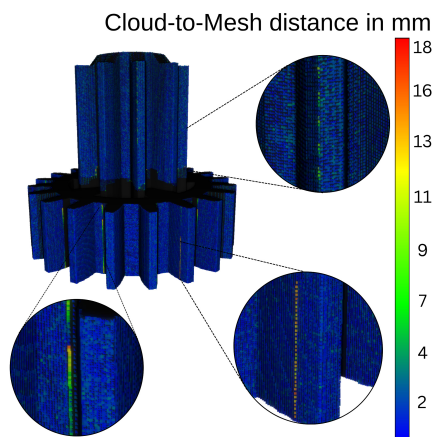


Figure 4. Mixed-pixel artefacts along object edges, caused by partial beam hits on multiple surfaces.

where surfaces are not visible from the scanner’s perspective (Figure 2). Finally, density variations occur due to differences in scan angle and distance, causing uneven point distributions across the object surface (Figure 5).

All provided point clouds underwent a rigid-body transformation, which serves as the ground truth for point cloud-to-model registration algorithms. The transformed point clouds, the original three-dimensional models, and the corresponding homogeneous transformation matrices are provided in the dataset. The transformations were applied as follows. For each object, a point cloud scan is simulated and then rotated around the x -, y -, and z -axes using independent random angles between 0° and 360° . The point cloud is then translated. Due to the fact that not every model has a true scale, the translation is conducted relatively. As a result, a maximum translation factor f with $5 \geq f \geq -5$ is defined. For each spatial direction (x, y, z) , a random value between the negative and positive maximum translation factor was selected and multiplied by the extent of the point cloud in that direction. This ensures that the applied translations are meaningful regardless of the scale of the point cloud. Furthermore, the point clouds were scaled with a chance of 50% by a scaling factor between 0.5 and 1.5.

The sources of the 3D models for the simulated point clouds range from publicly available datasets, including the ABC dataset (Koch et al., 2019),

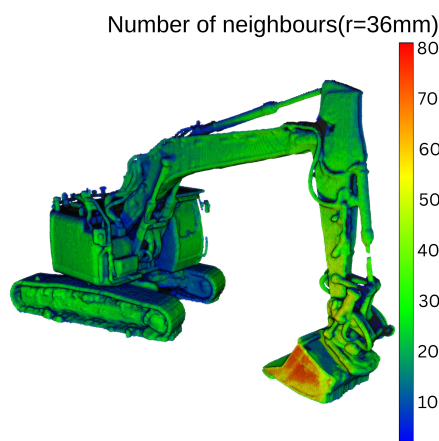


Figure 5. Point density variation across the surface; warmer colors indicate higher local density.

Parameter	Value
Accuracy	1.2 mm
Beam divergence	0.23 mrad
Maximum rotation speed	18° s^{-1}
Pulse frequencies	125,000 Hz
	250,000 Hz
	500,000 Hz
Pulse length	1,000,000 Hz
	0.75 ns
	0.4 m
Minimum scan range	0.4 m
Maximum vertical scan angle	145°
Wavelength	1,550 nm
Minimum scan frequency	4 Hz
Maximum scan frequency	100 Hz

Table 3. Specifications of the Leica ScanStation P40 laser scanner

ModelNet (Wu et al., 2015), Fusion360 (Willis et al., 2022), and Thingi10k (Zhou and Jacobson, 2016), to user-contributed, open-licensed 3D models hosted on the Sketchfab¹ platform. In addition, for real-world scans, the ISPRS indoor modelling dataset (Khoshelham et al., 2020) was included in the benchmark. All models were carefully curated to meet the diversity and quality requirements of the PC2Model dataset.

4. Simulation software and Blender add-on

To simulate the scanning of the 3D models, we selected Helios++ (Winiwarter et al., 2022) from several available simulation tools. Helios++ recreates laser-scanning environments with high fidelity, ensuring that the generated point clouds resemble real-world scans. The software is open source and supports the import of custom 3D models in Wavefront Object (OBJ) format. In combination with configurable simulation parameters and transparent file handling, these features enable realistic scan behaviour at a manageable computational cost. Helios++ supports simulations for terrestrial, airborne, and mobile laser scanners. In this project, terrestrial scanning was used. Among the available scanning methods, polygonal mirror deflection was selected. A key feature of Helios++ is its waveform-based laser beam simulation. Instead of modelling beams as straight lines, beam footprints are represented with a finite spatial extent, and reflected energy is integrated to derive surface characteristics. Beam divergence is modelled using multiple sub-rays sampled around a central ray, each of which is simulated individually, enabling control over parameters that affect scanning accuracy. Ray tracing is used to compute intersections between rays and scene geometry, and additional noise sources, including range measurement noise and sensor position uncertainty, are incorporated to generate more realistic point clouds. As Helios++ provides a high-fidelity laser scanner simulation, it was possible to recreate a scanner resembling the Leica ScanStation P40. The scanner specifications were taken from the datasheet² and implemented within Helios++ and are listed in Table 3. To achieve a uniform sampling density, the vertical and horizontal angular resolutions were set to the same value of 0.174° . Accordingly, the head rotation speed was set to 13.05 m s^{-1} , the pulse frequency to 125,000 Hz, and the scan frequency to 75 Hz.

The setup of the simulation environments includes precise placement, translation, and scaling of the 3D objects, as well

¹ <https://sketchfab.com>

² <https://leica-geosystems.com>

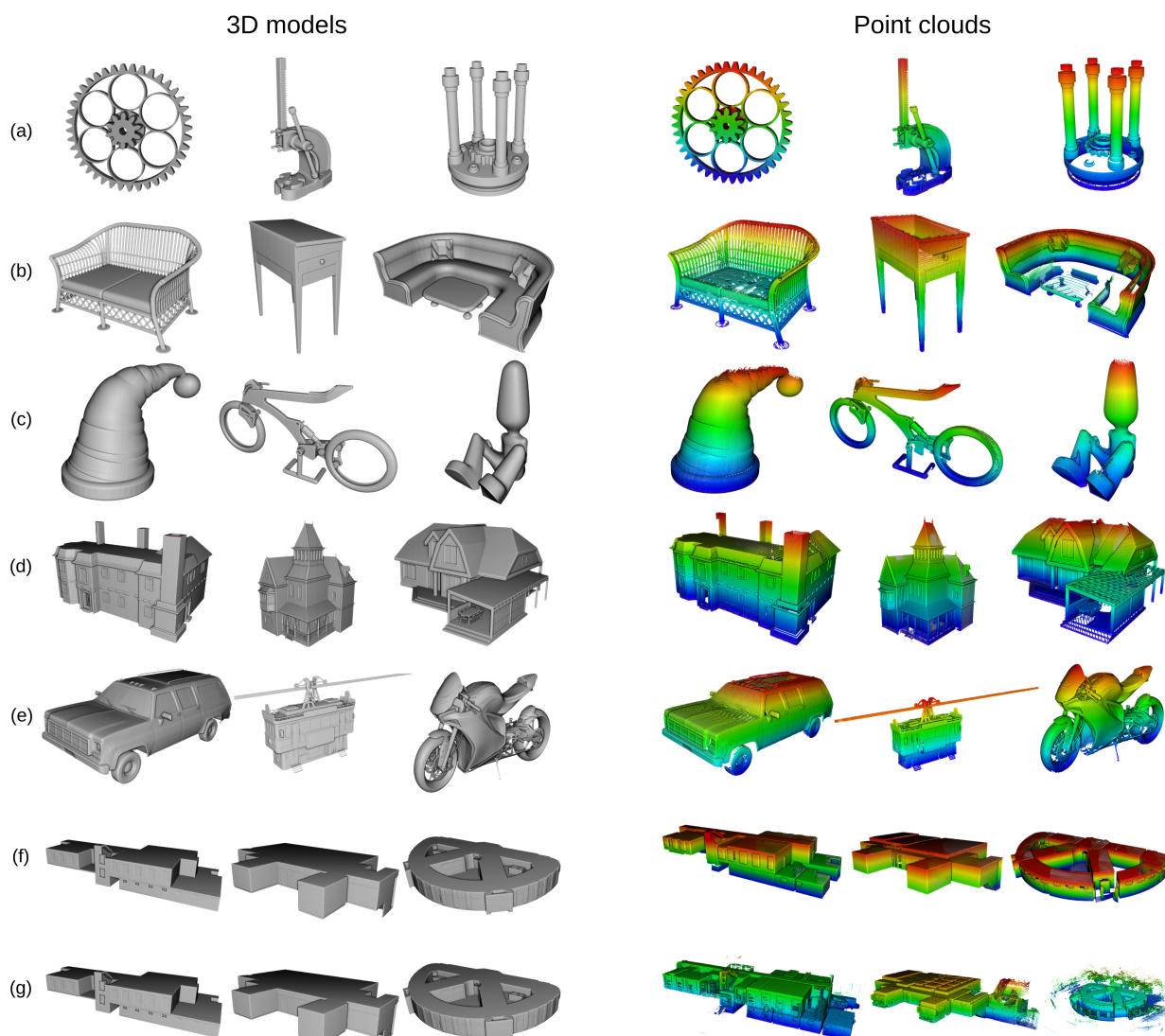


Figure 6. Representative dataset samples spanning seven categories: (a) mechanical objects, (b) furniture, (c) home décor, (d) houses, (e) vehicles, (f) indoor spaces, and (g) indoor spaces (real). Each sample is shown as a point cloud (right) alongside its corresponding 3D reference model (left).

as careful positioning of the laser scanner around, and when applicable within, the objects. For larger-scale datasets, a solution was required to make the setup process faster, more intuitive, and supported by a visual interface for verification. To address this, a Blender³ add-on was developed that allows users to configure 3D scanning environments for Helios++ directly within Blender's viewport. Using the add-on, scanners can be positioned accurately at suitable locations, and scanner settings can be adjusted while the resulting angular resolutions are displayed immediately. When a simulation is executed, the add-on automatically generates the scene and survey files in Extensible Markup Language (XML) format, which serve as the required inputs for Helios++. After the simulation is completed, the point clouds from each scanner are converted to ASTM E57 (E57) format, with the scanner location embedded in the files. These point clouds are unified into a single E57 file containing the XYZ coordinates and scanner positions, while individual scans remain distinguishable within the file. Finally, all output files are organised according to the PC2Model directory structure. The add-on is maintained on GitHub, which

also provides the files used to calculate the dataset statistics: <https://github.com/saidharb/PC2Model.git>

5. Evaluation strategy

To establish a consistent and rigorous evaluation strategy, we apply a known rigid body transformation to each CAD model. The parameters of this transformation serve as the ground truth targets for algorithms performing point cloud-to-model co-registration. We will use ICP as the baseline benchmark for our dataset for researchers to compare their results. For evaluation, we employ established metrics commonly used in related literature to assess both geometric alignment quality and spatial completeness. In addition, we evaluate the accuracy of the estimated rigid transformation to quantify co-registration performance.

Level of Accuracy (LOA): The geometric accuracy of the registration could be measured using the Level of Accuracy (LOA), which quantifies how closely the registered point cloud aligns with the reconstructed 3D model

³ <https://www.blender.org>

surface (Ersöz and Bosché, 2026). We employ two complementary distance-based measures to capture both overall alignment quality and robustness to outliers.

Let $\mathcal{P} = \{\mathbf{p}_i \in \mathbb{R}^3 \mid i = 1, \dots, N\}$ denote the registered point cloud and $\mathcal{S}$ represent the surface of the 3D model. For each point $\mathbf{p}_i$, the closest point $\mathbf{q}_i \in \mathcal{S}$ on the model surface by orthogonal projection is determined. The mean closest-point unsigned distance measures the average geometric deviation between the point cloud and the surface.

$$\text{LOA}_{\text{mean}} = \frac{1}{N} \sum_{i=1}^N \|\mathbf{p}_i - \mathbf{q}_i\|_2 \quad (1)$$

In addition to the mean distance, we compute the median signed orthogonal distance between the point cloud and the surface. Let d_i denote the signed distance from point $\mathbf{p}_i$ to the surface $\mathcal{S}$, where the sign is defined with respect to the local surface normal. This metric is more robust to outliers and allows the detection of systematic alignment biases.

$$\text{LOA}_{\text{median}} = \text{median}_{i=1, \dots, N} (d_i) \quad (2)$$

Level of Coverage (LOC): While LOA focuses on geometric accuracy, it does not capture whether the entire model surface is sufficiently represented by the point cloud. To assess spatial completeness, we employ the Level of Coverage (LOC) (Ersöz and Bosché, 2026).

A regular grid $\mathcal{G} = \{\mathbf{g}_j \in \mathcal{S} \mid j = 1, \dots, M\}$ is sampled on the model surface. A grid point $\mathbf{g}_j$ is considered covered if there exists at least one point $\mathbf{p}_i \in \mathcal{P}$ whose Euclidean distance to $\mathbf{g}_j$ is below a predefined threshold τ .

$$\|\mathbf{p}_i - \mathbf{g}_j\|_2 \leq \tau \quad (3)$$

The Level of Coverage is then defined as the ratio of covered grid points to the total number of sampled grid points, using the indicator function $\mathbb{I}(\cdot)$.

$$\text{LOC} = \frac{1}{M} \sum_{j=1}^M \mathbb{I} \left(\min_{\mathbf{p}_i \in \mathcal{P}} \|\mathbf{p}_i - \mathbf{g}_j\|_2 \leq \tau \right) \quad (4)$$

In order to provide deeper insights into the geometric quality and spatial completeness of the benchmark data for point-to-mesh registration across different object categories, the category-level distributions of LOA and LOC are visualised in Figures 7 and 8, respectively. It is worth mentioning that the same metrics could be used for the evaluation of point cloud to model co-registration algorithms.

As it is visible in Figure 7, all simulated categories (e.g., vehicles, houses, indoor, and home décor) achieve mean LOA values of better than 1 cm, demonstrating high geometric alignment accuracy. As expected indoor spaces (real) exhibit higher LOA values reflecting clutter, and higher registration error in real-world data.

Figure 8 presents the LOC distributions across categories for multiple distance thresholds (10 cm to 25 cm). As expected,

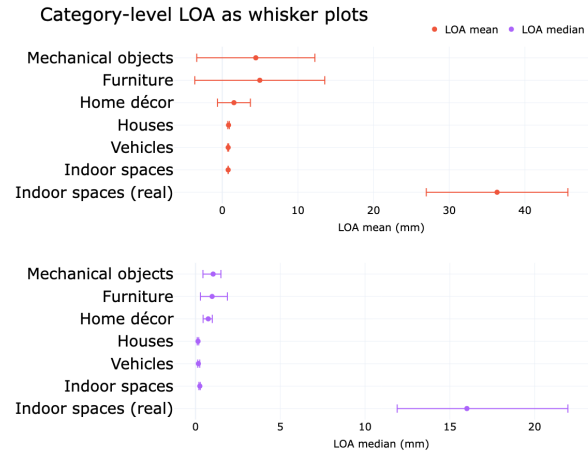


Figure 7. Category-level LOA of PC2Model objects.

LOC increases with higher threshold values for all categories. Simulated categories generally achieve high coverage, indicating that most of the model’s surfaces are well represented by the point cloud. However, categories like furniture show lower coverage at stricter thresholds, mainly due to occlusions. Indoor spaces (real) demonstrate comparatively lower LOC at smaller thresholds but improve significantly at larger thresholds.

Transformation Error Metrics: The accuracy of the estimated transformation is evaluated by comparing it to the ground truth using the error matrix $\mathbf{T}_{\text{err}} \in \text{SE}(3)$, defined as:

$$\mathbf{T}_{\text{err}} = \mathbf{T}_{\text{est}}^{-1} \mathbf{T}_{\text{gt}} = \begin{bmatrix} \mathbf{R}_{\text{err}} & \mathbf{t}_{\text{err}} \\ \mathbf{0}^\top & 1 \end{bmatrix} \quad (5)$$

where $\mathbf{T}_{\text{gt}}$ is the ground truth transformation from point cloud to model and $\mathbf{T}_{\text{est}}$ is the estimated transformation from point cloud to model. A perfect estimate yields $\mathbf{T}_{\text{err}} = \mathbf{I}$.

The *translation error* (e_t) is computed as the Euclidean norm of the translation vector. To make translation errors comparable across different point densities, they are divided by the length of their bounding box diagonal d :

$$e_t = \frac{\|\mathbf{t}_{\text{err}}\|_2}{d} \quad (6)$$

The *rotation error* (e_r) measures the difference between the estimated rotation and the ground truth

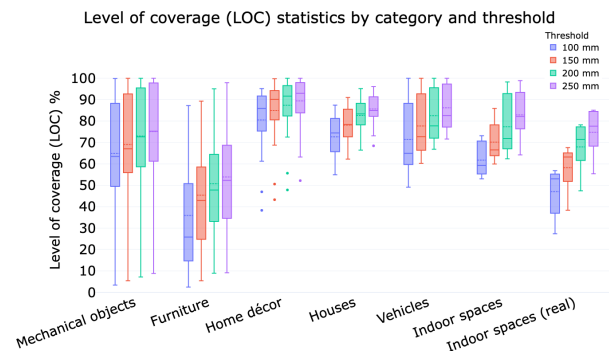


Figure 8. Level of Coverage (LOC) distributions across categories for different thresholds.

rotation (Hartley and Zisserman, 2004). Formally, it is defined as the *geodesic distance* on the rotation manifold $SO(3)$, which corresponds to the minimal angle of rotation needed to align the two rotations.

Let $\mathbf{R}_{\text{err}}$ denote the relative rotation between the estimated and ground truth rotations:

$$\mathbf{R}_{\text{err}} = \mathbf{R}_{\text{est}}^{-1} \mathbf{R}_{\text{gt}} \quad (7)$$

The rotation error e_r is then computed from the rotation matrix $\mathbf{R}_{\text{err}}$ as:

$$e_r = \arccos \left(\frac{\text{trace}(\mathbf{R}_{\text{err}}) - 1}{2} \right)$$

where $\text{trace}(\cdot)$ denotes the sum of the diagonal elements of a matrix.

Baseline evaluation: We use ICP to quantify the achievable alignment accuracy of a standard local registration method under controlled initial conditions. ICP is a local optimisation method and therefore requires a reasonably good initial alignment (rough co-registration). To obtain a meaningful baseline, we applied only a moderate transformation to the point clouds: translations were limited to at most 0.1 times the maximum bounding-box extent per axis, rotations were restricted to the range $[-10^\circ, 10^\circ]$, and no scaling was applied, in contrast to the much larger transformations in our final dataset. For ICP, a uniformly sampled point cloud of the 3D model serves as the reference. ICP is then performed to align the moderately transformed scan back to this reference. The ICP algorithm was applied using Open3D (Zhou et al., 2018) and CloudCompare⁴ with a point-to-plane error metric and a maximum of 2000 iterations. The convergence criterion was set to an RMS difference of 1.0×10^{-5} . To control computational cost and avoid bias from extremely dense scans, the transformed point cloud was randomly downsampled to at most 50,000 points prior to registration, preserving the global geometric structure while significantly reducing runtime and memory requirements. The results can be seen in Table 4.

Source	Category	Avg. e_t in mm	Avg. e_r in $^\circ$
Simulated from 3D model	Mechanical objects	4	1.2
	Furniture	1	0.0
	Home décor	2	0.5
	Houses	0	0.0
	Vehicles	0	0.0
	Indoor spaces	0	0.0
Real	Indoor spaces	5	1.0

Table 4. ICP alignment errors under coarse initialization: normalized translation error e_t and rotation error e_r .

The PC2Model dataset poses a range of algorithmic challenges for point cloud registration methods. First, the dataset contains a large variety of object categories and corresponding shapes, which prevents algorithms from being trained on a single object class and instead requires strong generalisation capabilities. This diversity extends beyond object categories to the properties of the point clouds themselves: object sizes vary significantly, point densities differ across scans, and different

scanner positions result in varying and incomplete views of the same object. Due to the simulation capabilities of Helios++, the generated point clouds exhibit a high degree of realism. Rather than idealised scans, the data contain real-world artefacts such as occlusions, partial scans, and point accuracy deviations. Point outliers, however, are a comparatively minor challenge, as objects are placed in empty scenes, making spurious measurements rare. Another major challenge lies in estimating the rigid transformation between point clouds. The transformations are randomly performed, leading to potentially large translations and rotations, such that initial misalignment can be substantial, although partial overlap may still be present. In addition, the dataset introduces significant computational challenges, as individual point clouds can contain up to 132 million points. Finally, even objects within the same category exhibit notable geometric variation, requiring registration algorithms to robustly account for diverse local and global features.

6. Conclusions

The PC2Model benchmark dataset provides a controlled, scalable, and cost-effective resource comprising point clouds and corresponding 3D models for training and evaluating both conventional and deep learning-based registration methods. By combining simulated point clouds with real-world scans, the dataset overcomes limitations commonly associated with exclusively real-world data, such as noise, occlusions, and limited availability. This benchmark is intended to support the research community in developing, comparing, and validating registration methods, contributing to more robust and accurate solutions in the field. The dataset is publicly accessible at: <https://zenodo.org/records/17581812>.

Future work will focus on extending the PC2Model dataset to increase its diversity and complexity for registration tasks. Planned enhancements include expanding object samples and categories, incorporating partial occlusions, and simulating varying noise levels to better reflect real-world scanning conditions. In addition, we plan to develop a web-based evaluation platform to facilitate the benchmarking of registration methods and visualisation of performance metrics.

Acknowledgments: This work is funded by the ISPRS Scientific Initiatives 2025. It is also supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation)– TRR 277/2 2024– Project number 414265976.

References

- Besl, P., McKay, N. D., 1992. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2), 239-256. <https://doi.org/10.1109/34.121791>.
- Biber, P., Strasser, W., 2003. The normal distributions transform: a new approach to laser scan matching. *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, 3, 2743–2748 vol.3. <https://doi.org/10.1109/IROS.2003.1249285>.
- Chen, X., Vizzo, I., Labe, T., Behley, J., Stachniss, C., 2021. Range image-based lidar localization for autonomous vehicles. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 5802–5808. <https://doi.org/10.1109/icra48506.2021.9561335>.

⁴ <https://www.cloudcompare.org>

- Ersöz, A. B., Bosché, F., 2026. Evaluating confidence in geometric matching between 3D point clouds and BIM models by integrating coverage, distance, and distribution metrics. *Automation in Construction*, 181, 106636. <https://doi.org/10.1016/j.autcon.2025.106636>.
- Hartley, R., Zisserman, A., 2004. *Multiple View Geometry in Computer Vision*. 2 edn, Cambridge University Press.
- Hinterstoisser, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G., Konolige, K., Navab, N., 2013. Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes. K. M. Lee, Y. Matsushita, J. M. Rehg, Z. Hu (eds), *Computer Vision – ACCV 2012*, Springer Berlin Heidelberg, Berlin, Heidelberg, 548–562. https://doi.org/10.1007/978-3-642-37331-2_42.
- Hodaň, T., Michel, F., Brachmann, E., Kehl, W., Buch, A. G., Kraft, D., Drost, B., Vidal, J., Ihrke, S., Zabulis, X., Sahin, C., Manhardt, F., Tombari, F., Kim, T.-K., Matas, J., Rother, C., 2018. Bop: Benchmark for 6D object pose estimation. V. Ferrari, M. Hebert, C. Sminchisescu, Y. Weiss (eds), *Computer Vision – ECCV 2018*, Springer International Publishing, Cham, 19–35. https://doi.org/10.1007/978-3-030-01249-6_2.
- Huang, Z., Wen, Y., Wang, Z., Ren, J., Jia, K., 2024. Surface Reconstruction From Point Clouds: A Survey and a Benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12), 9727–9748. <https://doi.org/10.1109/TPAMI.2024.3429209>.
- Khoshelham, K., Tran, H., Acharya, D., Vilariño, L. D., Kang, Z., Dalyot, S., 2020. The ISPRS Benchmark on Indoor Modelling – Preliminary Results. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLIII-B5-2020, 207–211. <https://doi.org/10.5194/isprs-archives-XLIII-B5-2020-207-2020>.
- Koch, S., Matveev, A., Jiang, Z., Williams, F., Artemov, A., Burnaev, E., Alexa, M., Zorin, D., Panozzo, D., 2019. ABC: A Big CAD Model Dataset for Geometric Deep Learning. 9593–9603. <https://doi.org/10.1109/CVPR.2019.00983>.
- Maboudi, M., Mawas, K., Gerke, M., 2025. Geometric Inspection in 3D Concrete Manufacturing: Comparison of Data Capturing Techniques. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLVIII-G-2025, 1029–1034. <https://doi.org/10.5194/isprs-archives-XLVIII-G-2025-1029-2025>.
- Mawas, K., Maboudi, M., Gerke, M., 2026. A review on geometry and surface inspection in 3D concrete printing. *Cement and Concrete Research*, 199, 108030. <https://doi.org/10.1016/j.cemconres.2025.108030>.
- Mock, A., Wiemann, T., Pütz, S., Hertzberg, J., 2024. Micp-l: Mesh-based icp for robot localization using hardware-accelerated ray casting. *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 10664–10671. <https://doi.org/10.1109/iro558592.2024.10802360>.
- Mora, R., Martín-Jiménez, J. A., Lagüela, S., González-Aguilera, D., 2021. Automatic Point-Cloud Registration for Quality Control in Building Works. *Applied Sciences*, 11(4). <https://doi.org/10.3390/app11041465>.
- Saiti, E., Theoharis, T., 2022. Multimodal registration across 3D point clouds and CT-volumes. *Computers & Graphics*, 106, 259–266. <https://doi.org/10.1016/j.cag.2022.06.012>.
- Sarode, V., Li, X., Goforth, H., Aoki, Y., Srivatsan, R. A., Lucey, S., Choset, H., 2019. Pcnnet: Point cloud registration network using pointnet encoding. <https://arxiv.org/abs/1908.07906>.
- Vizzo, I., Chen, X., Chebrolu, N., Behley, J., Stachniss, C., 2021. Poisson surface reconstruction for lidar odometry and mapping. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 5624–5630. <https://doi.org/10.1109/ICRA48506.2021.9562069>.
- Wang, Y., Solomon, J., 2019. Deep closest point: Learning representations for point cloud registration. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, IEEE Computer Society, Los Alamitos, CA, USA, 3522–3531. <https://doi.org/10.1109/ICCV.2019.00362>.
- Willis, K. D., Jayaraman, P. K., Chu, H., Tian, Y., Li, Y., Grandi, D., Sanghi, A., Tran, L., Lambourne, J. G., Solar-Lezama, A., Matusik, W., 2022. JoinABLE: Learning Bottom-Up Assembly of Parametric CAD Joints. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 15849–15860. <https://arxiv.org/abs/2111.12772>.
- Winiwarter, L., Esmorís Pena, A. M., Weiser, H., Anders, K., Martínez Sánchez, J., Searle, M., Höfle, B., 2022. Virtual laser scanning with HELIOS++: A novel take on ray tracing-based simulation of topographic full-waveform 3D laser scanning. *Remote Sensing of Environment*, 269, 112772. <https://doi.org/10.1016/j.rse.2021.112772>.
- Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J., 2015. 3D ShapeNets: A Deep Representation for Volumetric Shapes. 1912–1920. <https://doi.org/10.1109/CVPR.2015.7298801>.
- Yang, J., Li, H., Campbell, D., Jia, Y., 2016. Go-ICP: A Globally Optimal Solution to 3D ICP Point-Set Registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(11), 2241–2254. <https://doi.org/10.1109/TPAMI.2015.2513405>.
- Zhao, Y., Zhao, H., Radanović, M., Khoshelham, K., 2022. A Unified Framework for Automated Registration of Point Clouds, Mesh Surfaces and 3D Models by Using Planar Surfaces. *The Photogrammetric Record*, 37(180). <https://doi.org/10.1111/phor.12428>.
- Zhou, Q., Jacobson, A., 2016. Thingi10K: A Dataset of 10,000 3D-Printing Models. <https://doi.org/10.48550/arXiv.1605.04797>.
- Zhou, Q., Park, J., Koltun, V., 2018. Open3D: A Modern Library for 3D Data Processing. *CoRR*, abs/1801.09847. <http://arxiv.org/abs/1801.09847>.
- Zhou, Q.-Y., Park, J., Koltun, V., 2016. Fast global registration. B. Leibe, J. Matas, N. Sebe, M. Welling (eds), *Computer Vision – ECCV 2016*, Springer International Publishing, Cham, 766–782. https://doi.org/10.1007/978-3-319-46475-6_47.